Подключаем Git к Android Studio

Android Studio умеет работать с системами контроля версий (version control system, сокр.VCS). Самой популярной системой является Git, которая стала практически стандартом во многих языках программирования.

Сама по себе Git управляется через командную строку. Для изучения её возможностей есть множество документации. Мы сфокусируемся на подключении Git к проекту в Android Studio.

Чтобы лучше понять объяснения, откройте старый проект Неllo Kitty или создайте его заново, если успели его удалить.

Для начала нужно установить Git. Перейдите на страницу загрузки <http://git-scm.com/downloads> и скачайте последнюю версию под вашу операционную систему.

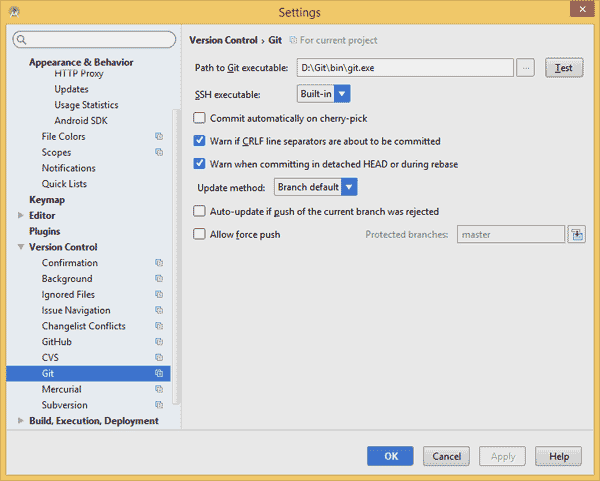
Запускаем процесс инсталяции. Не рекомендую устанавливать в папку по умолчанию в системной папке Windows **C:\Program Files**, так как из-за пробела между словами часто возникают проблемы у многих пользователей при работе с командной строкой. Я установил в папке **D:\Git**. Желательно также прописать путь к папке **D:\Git\bin\** в переменной окружения **PATH**.

Запускаем файл **git-bash.exe** для запуска консоли. Следует сконфигурировать Git, указав своё имя и электронный адрес, чтобы можно было установить авторство изменений в коде. Вводим по очереди две команды:

$ git config --global user.name "Ivan Ivanov"

$ git config --global user.email "iivanov@gmail.com"

Возвращаемся в студию. Выбираем в меню **File | Settings** и в диалоговом окне в левой части выбираем секцию **Version Control | Git**. Нажимаем кнопку с многоточием и находим нужный файл на диске.

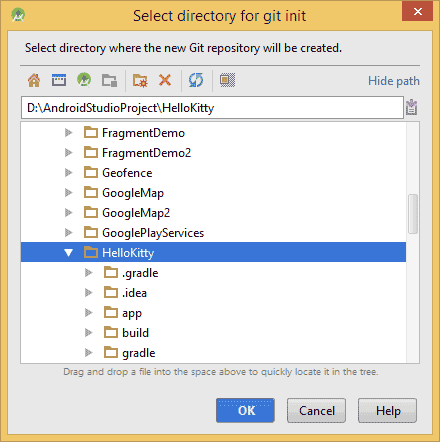


Для проверки можно щёлкнуть по кнопке **Test**, вы должны увидеть радостное сообщение в успешном случае. Закрываем окно настроек, применив изменения.



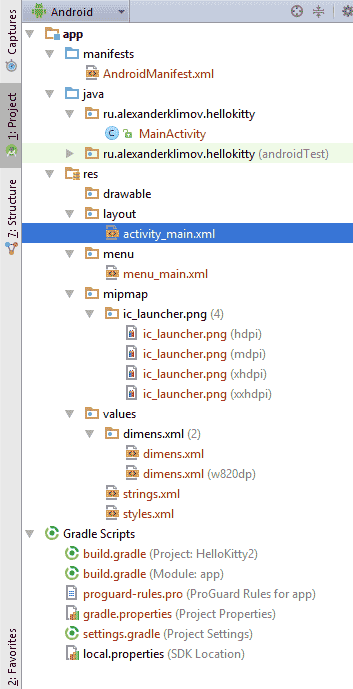
Данная настройка запоминается и в новых проектах этот шаг можно пропустить.

Далее идём в меню **VCS | Import into Version Control | Create Git Repository** и в диалоговом окне выбираем корневую папку проекта. Для удобства можно сразу нажать на значок студии (третий слева), чтобы сразу переместиться в корневую папку, если окно откроется с другой папкой.



Нажимаем кнопку **OK** и создаём локальный Git-репозиторий.

Как только вы это сделаете, в студии произойдут удивительные изменения. Многие имена файлов в левой панели окрасятся в коричневый цвет.



Коричневый цвет шрифта означает, что файл распознан системой контроля версий на локальном компьютере, но ещё не добавлен в репозиторий. Нам следует подсказать системе об этом.

Но не будем торопиться. При создании локального репозитория студия также создала несколько специальных файлов **.gitignore**, которые помогают системе контроля версий игнорировать некоторые файлы проекта при изменениях. Один такой файл вы можете найти в корневой папке проекта, а второй в папке **app**. Можете открыть их любым текстовым редактором. Начнём с файла из корневой папки.

.gradle

/local.properties

/.idea/workspace.xml

/.idea/libraries

.DS\_Store

/build

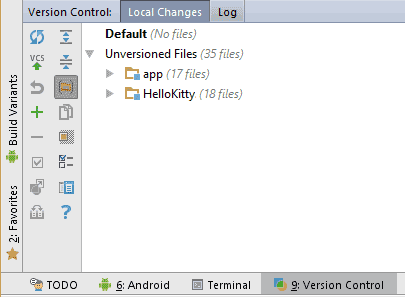
/captures

Как видим, Git будет игнорировать файл **.idea/workspace.xml**, который относится к конфигурации самой студии на вашем компьютере и к самому проекту не имеет отношения. Аналогично будет проигнорирован файл **local.properties**, который является уникальным для каждого компьютера. Можно указывать не только отдельные файлы, но и папки. Например, в списке присутствует папка **/build**. В эту папку попадают файлы при компиляции. Их нет смысла отслеживать, так как они постоянно меняются, когда вы запускаете приложение для проверки. Все файлы, которые должны быть проигнорированы, выводятся обычным чёрным цветом. Обратите внимание на имя файла **local.properties** на скриншоте выше.

Кроме чёрного и коричневого цвета, имена файлов могут также выводиться синим цветом (файл изменён), зелёным (новый файл).

При подключении Git в нижней части студии появится новая вкладка **Version Control**. Откройте её. Сейчас вы видите две секции: **Default** и **Unversioned Files**. Секция **Default** сейчас пуста. При изменении или создании новых файлов они попадут в эту секцию. Секция **Unversioned Files** содержит файлы, которые ещё не были учтены системой контроля версий.

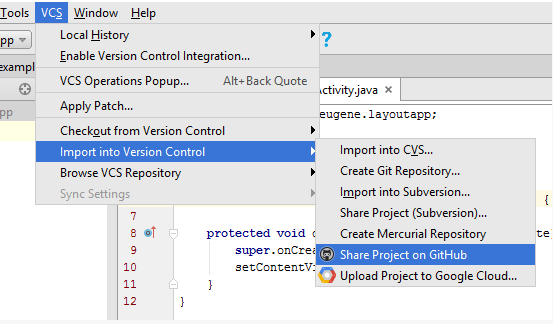
При создании нового проекта файлы автоматически не учитываются и находятся в секции **Unversioned Files**. Мы хотим их перенести в репозиторий. В левой части панели находятся две колонки со значками. Найдите значок с изображением папки в скобках (при подведении курсора появится подсказка **Group by Directory**) и нажмите на неё. Файлы будут сгруппированы, как в проекте. Так проще вам будет понять структуру.



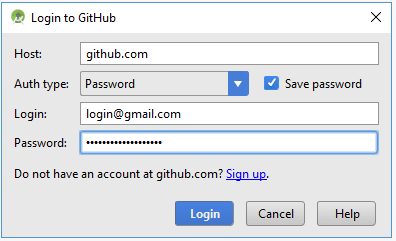
Щёлкаем правой кнопкой мыши на **Unversioned Files** и выбираем в контекстном меню **Add to VCS**. Либо можно перетащить мышкой эту секцию на секцию **Default**. В результате все файлы переместятся и будут учтены системой контроля версий.

После добавляения файлов нажмите на значок с зелёной стрелкой вверх и надписью VCS (**Commit Changes**). Откроется диалоговое окно. В текстовом поле **Commit Message** введите текст, поясняющий изменения в проекте и нажмите кнопку **Commit**.

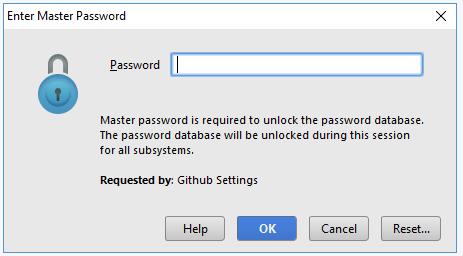
Возьмем какой-нибудь проект. В Android Studio перейдем к пункту меню **VCS -> Import into Version Control -> Share Project on Github**.



Далее надо будет ввести логин и пароль от учетной записи на github.com:

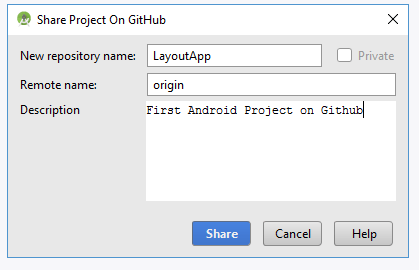


Затем будет предложено ввести дополнительный пароль (Master password):



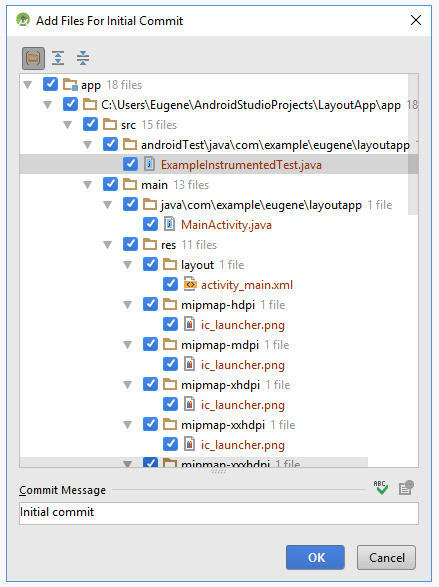
Если подобного пароля не указывалось ранее, оставим это полу пустым и нажмем OK.

Далее будет предложено указать имя репозитория, удаленное имя и описание:



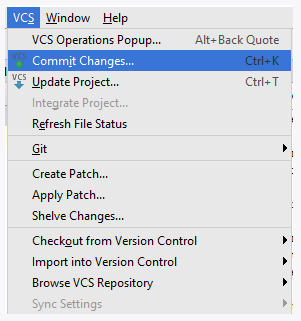
Описание лучше писать на английском. После ввода всех данных нажмем на кнопку Share.

После этого будет предложено указать те файлы проекта, которые мы хотим закинуть на github:

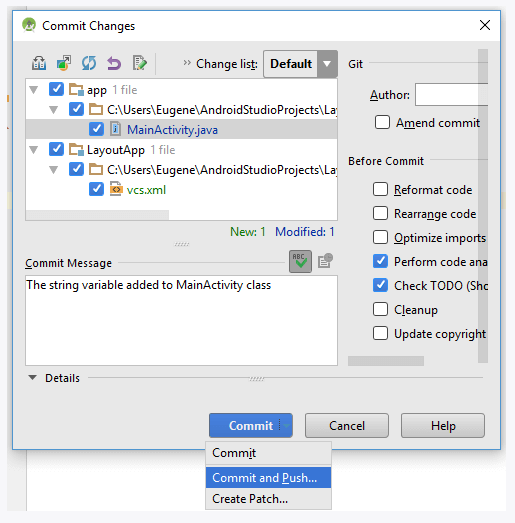


После удачного экспорта файлов в Android Studio отобразится соответствующее сообщение.

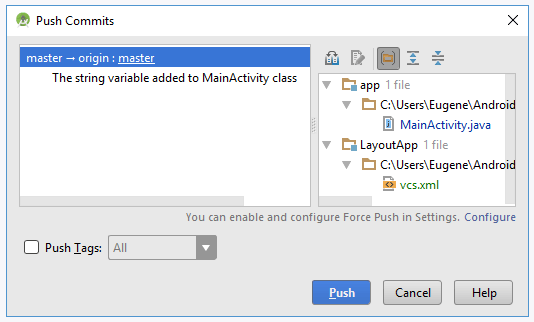
Итак, наш проект уже расположен на github, однако впоследствии мы можем произвести в нем изменения, например, добавить файлы кода, изменить какой-то код. И в этом случае нам надо будет обновить проект в репозитории. Для этого перейдем к пункту меню **VCS -> Commit Changes...**:



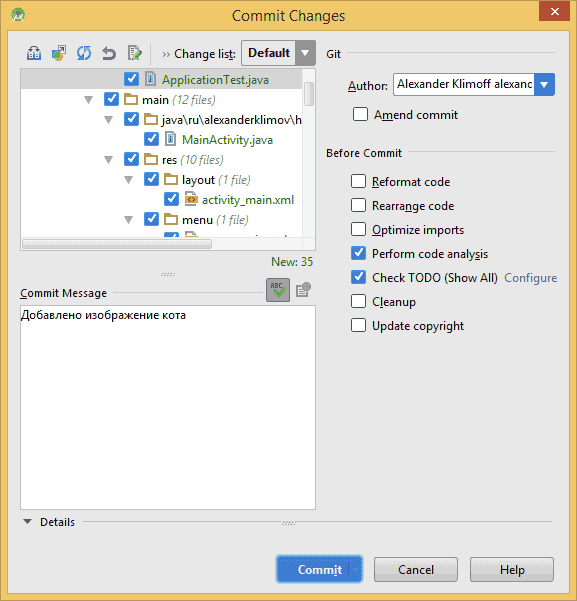
После этого отобразится окошко со сделанными изменениями со времени последнего коммита. В поле Commit Message укажем хаактер сделанных изменений. Для завершения обновления нажмем внизу окошка на кнопку **Commit and Push...**:



Затем отобразиться дополнительное окошко, где надо подтвердить коммит. Для этого нажмем на кнопку **Push**:



И через некоторое время обновления будут применены к репозиторию на github.com.



Файлы исчезнут из секции **Default** и теперь находятся под контролем Git. При изменении файл снова попадёт в данную секцию и вам снова нужно выполнить предыдущую операцию **Commit**.

Например, откроем файл манифеста и добавим разрешение на работу с интернетом. Файл окрасится в синий цвет. Комментируем изменения в проекте, добавив сообщение **Добавлено разрешение на интернет.**

Просматривать изменения можно на вкладке **Log**

Таким образом мы познакомились с базовыми приёмами работы с Git.